AD

RE-TR-71-72

A DIGITAL COMPUTER PROGRAM TO DETERMINE THE TWO-DIMENSIONAL TEMPERATURE PROFILE IN GUN TUBES

TECHNICAL REPORT

Dr. William J. Leech

and

George E. Stiles

February 1972

RESEARCH DIRECTORATE

WEAPONS LABORATORY AT ROCK ISLAND

RESEARCH, DEVELOPMENT AND ENGINEERING DIRECTORATE

U. S. ARMY WEAPONS COMMAND

Approved for public release, distribution unlimited.

DISPOSITION INSTRUCTIONS:

Destroy this report when it is no longer needed. Do not return it to the originator.

DISCLAIMER:

The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

ACCESSION for

NTIS White Section ☑

DDC Buff Section ☐

UNANNOUNCED ☐

JUSTIFICATION ....

BY ....

DISTRIBUTION/AVAILABILITY CODES

DIST.	AVAIL. and/or SPECIAL
A	

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) U. S. Army Weapons Command Research, Dev. and Fng. Directorate Rock Island, Illinois 61201	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A DIGITAL COMPUTER PROGRAM TO DETERMINE THE TWO-DIMENSIONAL TEMPERATURE PROFILE IN GUN TUBES (U)

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

William J. Leech and George E. Stiles

6. REPORT DATE February 1972	7a. TOTAL NO. OF PAGES 40	7b. NO. OF REFS 2

8a. CONTRACT OR GRANT NO. b. PROJECT NO. DA 1W562604A607 c. AMS Code 552D.11.80700.01 d.	9a. ORIGINATOR'S REPORT NUMBER(S) RE-TR-71-72 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release, distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY U. S. Army Weapons Command

13. ABSTRACT

This study was undertaken by the Research Directorate, Weapons Laboratory at Rock Island, to develop a digital computer program by which the two-dimensional temperature profile in gun tubes can be computed under realistic physical conditions. A mathematical model was presented in which variable geometry, temperature-dependent thermal properties, and variable conditions at the boundaries are considered. A numerical algorithm, in which the method of explicit finite-differences is used, was developed for the mathematical model and was programmed for the digital computer. A numerical example was computed to check the computer program. The program and all subroutines functioned properly. No numerical instability nor convergence problems were encountered. (U) (Leech, W. J. and Stiles, G. E.)

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified
Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
1. Heat Conduction						
2. Temperature Distribution						
3. Finite Differences						
4. Gun Tubes						

Unclassified
Security Classification

AD

RESEARCH DIRECTORATE

WEAPONS LABORATORY AT ROCK ISLAND

RESEARCH, DEVELOPMENT AND ENGINEERING DIRECTORATE

U. S. ARMY WEAPONS COMMAND

TECHNICAL REPORT

RE-TR-71-72

A DIGITAL COMPUTER PROGRAM TO DETERMINE THE TWO-DIMENSIONAL TEMPERATURE PROFILE IN GUN TUBES

Dr. William J. Leech

and

George E. Stiles

February 1972

DA 1W562604A607 AMS Code 552D 11 80700 01

Approved for public release, distribution unlimited.

ABSTRACT

This study was undertaken by the Research Directorate, Weapons Laboratory at Rock Island, to develop a digital computer program by which the two-dimensional temperature profile in gun tubes can be computed under realistic physical conditions. A mathematical model was presented in which variable geometry, temperature-dependent thermal properties, and variable conditions at the boundaries are considered. A numerical algorithm, in which the method of explicit finite-differences is used, was developed for the mathematical model and was programmed for the digital computer. A numerical example was computed to check the computer program. The program and all subroutines functioned properly. No numerical instability nor convergence problems were encountered.

CONTENTS

	Page
Title Page	i
Abstract	ii
Contents	iii
Introduction	1
Mathematical Model	2
Numerical Algorithm	6
Description of Computer Program	12
Numerical Example	14
Summary	17
Literature Cited	18
Appendix A - Digital Computer Program	19
Appendix B - Flow Chart of Main Program	32
List of Symbols Used in Text	34
Distribution	35
DD Form 1473 (Document Control Data - R&D)	39

INTRODUCTION

Small caliber automatic weapons are subjected to extremely high operating pressures and temperatures. Energy from the hot propellant gas is absorbed by the gun tube at a much faster rate than it is dissipated to the surroundings. Temperature rises occur quite rapidly, and result in erosion or loss of strength of the gun tube material. High pressures may cause the tube to become ruptured when the temperatures are increased sufficiently. The important point is, therefore, that gun tube designers be able to predict the temperature distribution of a particular gun tube design. The purpose of the present work is to develop a digital computer program by which gun tube temperatures can be computed for physically realistic conditions. These physical conditions involve variable axial geometry, temperature-dependent thermal properties, variable firing schedules, and variable thermal boundary conditions at both the bore and exterior surfaces. Results obtained from the computer analyses may be used to determine areas of excessive temperature rise, to estimate maximum burst time, to provide information necessary for thermal stress analyses, and to indicate necessary changes for improved thermal performance.

MATHEMATICAL MODEL

In this section, the physical mechanisms of heat transfer from the propellant gas to the gun tube are described and a mathematical formulation is given for the temperature distribution in the tube. An illustration of the gun tube is shown in Figure 1.

The gun tube material is considered to be isotropic, but the thermal properties, $\rho(T)$, $C(T)$, and $K(T)$, are known functions of temperature. The assumption is that angular temperature variations are small, compared with radial and axial temperature variations. Thus, only a two-dimensional temperature field must be considered. Heat flows from the hot propellant gas, whose temperature is represented by $Tg(r,z,t)$, to the tube, whose temperature is denoted by $T(r,z,t)$. The assumption in this analysis is that the heat flux from the gas to the bore surface is specified or that the propellant gas temperature is a known function of time and position, and that a heat transfer coefficient, $h_1(R_1,z,t,T)$ exists which is also a known function. And finally, the assumption is that continuous variations in the outside diameter of the tube may be adequately approximated by a finite number of step changes in the exterior diameter. The actual diameter as being approximated by three step changes is shown in Figure 1. The number of step changes may be greater or less than three, dependent upon the situation. Close approximation of any taper of the outside diameter by use of a greater number of step changes is possible. The analysis will be illustrated with the use of three step changes. However, the computer program was written so that any desired number of step changes in the external diameter could be handled.

The governing partial differential equation for the gun tube is given by

$$\frac{\partial T}{\partial t} = \frac{K(T)}{\rho(T)C(T)}\left[\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial Z^2}\right]$$

$$+ \frac{1}{\rho(T)C(T)}\frac{\partial K}{\partial T}\left[\left(\frac{\partial T}{\partial r}\right)^2 + \left(\frac{\partial T}{\partial Z}\right)^2\right] \qquad (1)$$

Equation 1 is nonlinear due to the presence of temperature-dependent properties. The boundary conditions for the tube illustrated in Figure 1 are

$$r = R_1,\ 0 \le Z \le Z_3$$

$$-K(T)\frac{\partial T}{\partial r}(R_1,Z,t) = q''(R_1,Z,t) \qquad (2)$$

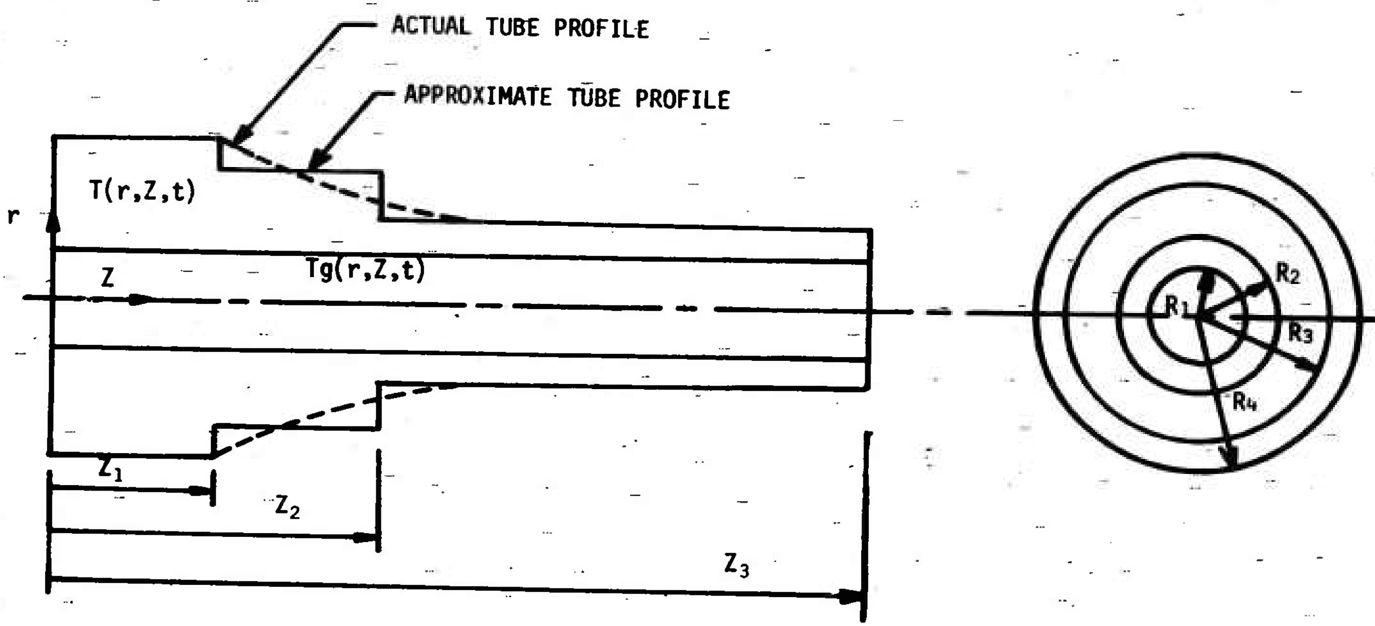


FIGURE 1 Illustration of Gun Tube

$r = R_4,\ 0 \le Z \le Z_1$

$$-K(T)\frac{\partial T}{\partial r}(R_4,Z,t) = h(T)[T(R_4,Z,t) - T_o] + \varepsilon(T)\sigma[T^4(R_4,Z,t) - T_o^4] \tag{3}$$

$r = R_3,\ Z_1 \le Z \le Z_2$

$$-K(T)\frac{\partial T}{\partial r}(R_3,Z,t) = h(T)[T(R_3,Z,t) - T_o] + \varepsilon(T)\sigma[T^4(R_3,Z,t) - T_o^4] \tag{4}$$

$r = R_2,\ Z_2 \le Z \le Z_3$

$$-K(T)\frac{\partial T}{\partial r}(R_2,Z,t) = h(T)[T(R_2,Z,t) - T_o] + \varepsilon(T)\sigma[T^4(R_2,Z,t) - T_o^4] \tag{5}$$

$0 < R < R_4,\ Z = 0$

$$-K(T)\frac{\partial T}{\partial Z}(r,0,t) = q''(r,0,t) \tag{6}$$

$R_3 \le r \le R_4,\ Z = Z_1$

$$-K(T)\frac{\partial T}{\partial Z}(r,Z_1,t) = 0 \tag{7}$$

$R_2 \le r \le R_3,\ Z = Z_2$

$$-K(T)\frac{\partial T}{\partial Z}(r,Z_2,t) = 0 \tag{8}$$

$R_1 \le r \le R_2,\ Z = Z_3$

$$-K(T)\frac{\partial T}{\partial Z}(r,Z_3,t) = h(T)[T(r,Z_3,t) - T_o] + \varepsilon(T)\sigma[T^4(r,Z_3,t) - T_o^4] \tag{9}$$

The initial conditions are

$$T(r,Z,0) = Ti\ (r,Z) \tag{10}$$

The heat flux given in Equation 2 may be a specified function, or may be expressed in terms of a heat transfer coefficient and the local difference between the bore surface temperature and the bore center line gas temperature. In the latter case, the boundary condition is given by

$$q''(R_1,Z,t) = h(R_1,Z,t)[Tg(0,Z,t) - T(R_1,Z,t)] \tag{11}$$

The heat flux given in Equation 6 must also be specified. This surface may lose energy to the surroundings, or exchange heat with some other section of the weapon. Boundary conditions for this surface must be specified on an individual basis.

The temperature-dependent thermal properties given in all the governing equations must be evaluated at the temperature of the point at which the equations are being evaluated. The heat fluxes at the boundary locations at which step changes have been used to approximate continuous variations in external diameter are assumed to be in the radial direction only. This is indicated in Equations 7 and 8. The radiation form factor for all other external surfaces has been taken as unity

The set of equations given above cannot be solved analytically, so numerical techniques must be employed. The method of explicit finite differences was chosen to solve the equations. The details of the numerical algorithm are given in the following section

NUMERICAL ALGORITHM

To determine the temperature distribution in the gun tube, the tube is first subdivided into a finite number of discrete lumps. The subdivision of the gun tube is illustrated in Figure 2. The tube has been divided into three sections in both the axial and the radial directions. The number of nodes in the first radial section is i_1, including the interface between sections 1 and 2. The second and third radial sections contain i_2 and i_3 nodes, respectively. In the first axial section, j_1 nodes are present, including the interface node. The second and the third axial sections contain j_2 and j_3 nodes, respectively. The spatial increments between the nodes are given by

$$\Delta r_1 = \frac{(R_2-R_1)}{(i_1-1)} \tag{12}$$

$$\Delta r_2 = \frac{(R_3-R_2)}{i_2} \tag{13}$$

$$\Delta r_3 = \frac{(R_4-R_3)}{i_3} \tag{14}$$

$$\Delta Z_1 = \frac{Z_1}{(j_1-1)} \tag{15}$$

$$\Delta Z_2 = \frac{(Z_2-Z_1)}{j_2} \tag{16}$$

$$\Delta Z_3 = \frac{(Z_3-Z_2)}{j_3} \tag{17}$$

The nodal point locations are

$$r_n = R_1 + \frac{(n-1)}{(i_1-1)}(R_2-R_1),\ 1 \le n \le i_1 \tag{18}$$

$$r_n = R_2 + \frac{(n-i_1)}{i_2}(R_3-R_2),\ i_1 < n \le i_1+i_2 \tag{19}$$

$$r_n = R_3 + \frac{(n-i_1-i_2)}{i_3}(R_4-R_3),\ i_1+i_2 < n \le i_1+i_2+i_3 \tag{20}$$

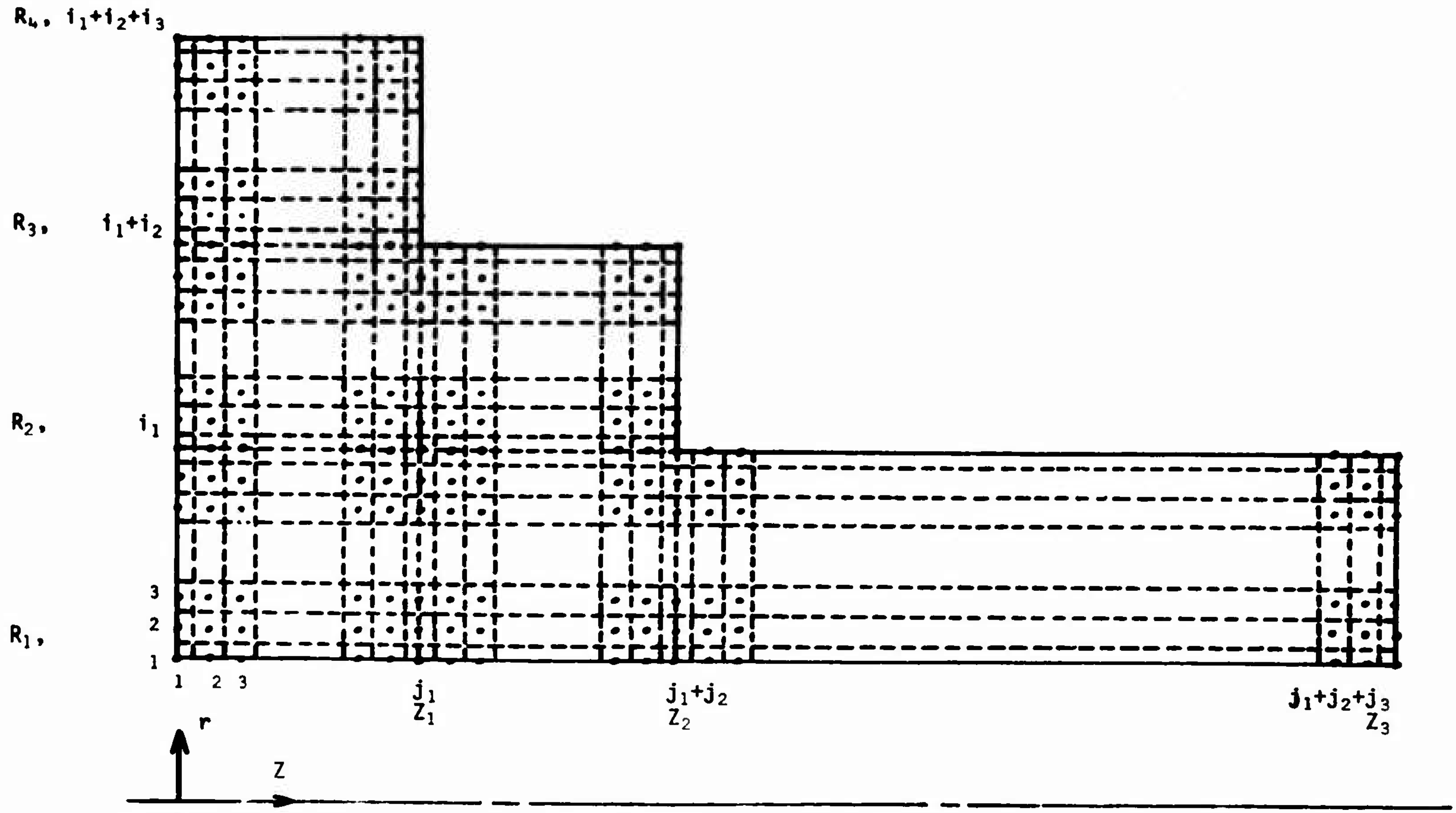


FIGURE 2 FINITE DIFFERENCE GRID NETWORK

$$Z_m = \frac{(m-1)}{(j_1-1)} Z_1, \; 1 \le m \le j_1 \tag{21}$$

$$Z_m = Z_1 + \frac{(m-j_1)}{j_2} (Z_2-Z_1), \; j_1 < m \le j_1+j_2 \tag{22}$$

$$Z_m = Z_2 + \frac{(m-j_1-j_2)}{j_3} (Z_3-Z_2), \; j_1+j_2 < m \le j_1+j_2+j_3 \tag{23}$$

The temperature of each lump is assumed to be uniform and equal to the temperature of its center or nodal point. The spatial derivatives appearing in the governing equations are approximated by finite difference relations, which have been determined from the simultaneous solutions of truncated Taylor series expansions. Equation 1, written in discrete difference notation, is

$$\left(\frac{\partial T}{\partial t}\right)_{n,m} = \frac{K_{n,m}}{\rho_{n,m} C_{n,m}} \left[\left(\frac{\partial^2 T}{\partial r}\right)_{n,m} + \frac{1}{r_n}\left(\frac{\partial T}{\partial r}\right)_{n,m}\right]$$

$$+ \frac{1}{\rho_{n,m} C_{n,m}} \left[\left(\frac{\partial T}{\partial r}\right)^2_{n,m} + \left(\frac{\partial T}{\partial Z}\right)^2_{n,m}\right] \tag{24}$$

Subscript n denotes the radial node location and subscript m denotes the axial node locations. The subscripts of the property values denote that they are evaluated at the temperature of node n,m.

$$K_{n,m} = K\,(T_{n,m}) \tag{25}$$

The boundary conditions must also be written in difference form. For example, Equation 2 is written as

$$-K_{1,m} \left(\frac{\partial T}{\partial r}\right)_{1,m} = q''_{1,m} \tag{26}$$

The spatial derivatives in Equations 24 and 26 are approximated by finite difference relations. A set of explicit algebraic equations result for the time rate of temperature change at each node. These rates of change are multiplied by a finite time increment to find the temperature at one increment of time later. The procedure is repeated until the final time period of interest has been reached. The finite difference expressions used to approximate the spatial derivatives are shown in Tables I and II. The particular derivatives shown are for the

radial direction. The expressions for the axial direction are of the same form, where the spatial increment is ΔZ, the subscript n is fixed, and the subscript m is a variable. The appropriate expression to use depends on the node of interest. For interior nodes, Number 2 in Table I and Number 3 in Table II are used. At the bore surface, Number 1 of Table II is used for the second derivative; the first derivative is determined from the boundary condition. The correct expressions for the external boundary nodes are those of the heat flux boundary conditions and Number 4 of Table II. The derivatives for the nodes adjacent to the boundaries are given by Numbers 2 and 3 of Table I and Number 2 of Table II. At the interface nodes, by which sections are separated in which the increments between nodes may become changed in size, the correct expressions are Numbers 4 and 5 of Table I and Table II, respectively.

The maximum time increment, by which numerical stability is ensured for the linear diffusion equation with this algorithm, is given by

$$\Delta t \leq \rho C \frac{\Delta X^2}{Kw} \tag{27}$$

where ΔX is the distance between nodal points and w is

$$w = 4 + 4 \frac{h\Delta X}{K} \tag{28}$$

The present set of equations are nonlinear and the maximum allowable time interval would be expected to be less than that given by Equation 27.

The sufficient condition for numerical stability and convergence is that both the first and the second laws of thermodynamics be satisfied. The satisfaction of the first law was verified by the performance of an energy balance on the tube after each time interval. No attempt was made to check the satisfaction of the second law. The belief was that the satisfaction of the first law provided an adequate check of numerical stability and convergence.

The explicit finite difference algorithm, described in this section, was programmed for the digital computer. The computer program is described in the following section, and a program listing is given in Appendix A.

TABLE I

FINITE DIFFERENCE EXPRESSIONS FOR FIRST DERIVATIVES

No.	$(\partial T/\partial r)_{n,m}$
1	$\frac{1}{6\Delta r}[-2T_{n-1,m} - 3T_{n,m} + 6T_{n+1,m} - T_{n+2,m}]$
2	$\frac{1}{12\Delta r}[T_{n-2,m} - 8T_{n-1,m} + 8T_{n+1,m} - T_{n+2,m}]$
3	$\frac{1}{6\Delta r}[T_{n-2,m} - 6T_{n-1,m} + 3T_{n,m} + 2T_{n+1,m}]$
4	$\frac{1}{\Delta r_2(1+\frac{\Delta r_2}{\Delta r_1})}[T_{n+1,m} - T_n(1 - \frac{\Delta r_2^2}{\Delta r_1^2}) - \frac{\Delta r_2^2}{\Delta r_1^2}T_{n-1,m}]$

TABLE II

FINITE DIFFERENCE EXPRESSIONS FOR SECOND DERIVATIVES

No.	$(\partial^2 T/\partial r^2)_{n,m}$
1	$\frac{1}{18\Delta r^2}[-85T_{n,m} + 108T_{n+1,m} - 27T_{n+2,m} + 4T_{n+3,m}] - \frac{11}{3\Delta r}(\frac{\partial r}{\partial r})_{n,m}$
2	$\frac{1}{\Delta r^2}[T_{n+1,m} - 2T_{n,m} + T_{n-1,m}]$
3	$\frac{1}{12\Delta r^2}[-T_{n-2,m} + 16T_{n-1,m} - 30T_{n,m} + 16T_{n+1,m} - T_{n+2,m}]$
4	$\frac{1}{18\Delta r^2}[-85T_{n,m} + 108T_{n-1,m} - 27T_{n-2,m} + 4T_{n-3,m}] + \frac{11}{3\Delta r}(\frac{\partial T}{\partial r})_{n,m}$
5	$\frac{2}{\Delta r_1 \Delta r_2 + \Delta r_2{}^2}[(\frac{\Delta r_2}{\Delta r_1}) T_{n-1,m} - (\frac{\Delta r_2}{\Delta r_1} + 1) T_{n,m} + T_{n+1,m}]$

DESCRIPTION OF COMPUTER PROGRAM

A digital computer program was written for the evaluation of the numerical algorithm described in the previous section. The computer program comprises a main program and eight subroutines. The main program contains the input, the output, the logic operations, and the computation operations for temperature changes. Detailed calculations are performed in the subroutines. The name and the purpose of each subroutine is given below:

1. CONV - Provides external convection coefficients and emissivities.
2. QZSUB - Contains operations to compute the axial heat fluxes at the external surfaces due to both radiation and convection.
3. QRSUB - Contains the operations necessary to compute the radial heat fluxes at the surfaces due to both radiation and convection.
4. AXIDER - Specifies the operations for the computation of the spatial derivatives in the axial direction.
5. RADDER - Provides computations for the spatial derivatives in the radial direction.
6. DKDT - Gives derivative of thermal conductivity with respect to time.
7. XKKS - Gives the calculation of thermal conductivity as a function of temperature.
8. LINEAR - Gives specific heat as a function of temperature.

The input to the digital computer program consists of seven READ statements whose required input data are listed below:

1. M - number of radial segments
   N - number of axial segments
   SIGMA - radiation coefficient
   TS - ambient temperature
   TTIME - termination time
   NP - iteration number at which printout is desired
2. JJS(I) - number of radial nodes in each segment
3. LLS(I) - number of axial nodes in each segment

4. RS(J) - radial boundaries of each segment

5. ZS(I) - axial boundaries of each segment

6. KRA(I) - number of radial segments in each axial segment

7. X(I), Y(I) - temperature versus specific heat data

Thermal property data in the subroutines are for SAE 4130 steel. If a different barrel material is to be analyzed, the functional relationships in these subroutines must be changed. The emissivities and external convection coefficients in the present subroutines are constant. If these values are not constant for any case being investigated, the proper functional relations must be added to the subroutines.

The output from the program consists of the time and the temperature at each node for those iterations for which printout is desired. A complete listing of the digital computer program, along with typical input and output data, is given in Appendix A.

NUMERICAL EXAMPLE

A numerical example was computed to check the digital computer program. The sole purpose of computing the numerical example was to ensure that the program and the subroutines were functioning properly. No specific weapon was considered. With reference to Figure 1, the geometric dimensions used in the example were

$$R_1 = 0.625 \text{ inch} \tag{29}$$

$$R_2 = 0.845 \text{ inch} \tag{30}$$

$$R_3 = 0.940 \text{ inch} \tag{31}$$

$$R_4 = 1.088 \text{ inch} \tag{32}$$

$$Z_1 = 3.16 \text{ inch} \tag{33}$$

$$Z_2 = 12.5 \text{ inch} \tag{34}$$

$$Z_3 = 42.0 \text{ inch} \tag{35}$$

Thermal property data for SAE 4130 steel were obtained from Figure 2.013, of Aerospace Structural Metals Handbook.[2] The data were adjusted to the curves, which are given below.

$$K = (28.3 - 0.0087T) \frac{\text{BTU}}{\text{hr ft °F}}, \; T \leq 1420\text{°F} \tag{36}$$

and

$$K = (10.39 + 0.00347T) \frac{\text{BTU}}{\text{hr ft °F}}, \; T > 1420\text{°F} \tag{37}$$

The density variations for SAE 4130 steel are small, and the following mean value of density was used.

$$\rho = 490 \text{ lb/ft}^3 \tag{38}$$

Tabular data for specific heat were used in conjunction with a linear interpolation subroutine. The specific heat data from Figure 2.015 of Aerospace Structural Metals Handbook[2] are given below:

T, °F	C, BTU/lb °F
0	0.108
200	0.112
400	0.125
600	0.132
800	0.150
1000	0.160
1200	0.185
1600	0.180
2000	0.180
2200	0.150

The temperature of the surroundings was constant and equal to 70°F. A constant convection coefficient of 5 BTU/hr ft °F and a constant emissivity of 0.5 were prescribed at the external boundaries. An effective mean propellant gas temperature of 2000°F and an effective mean heat transfer coefficient at the bore surface of 200 BTU/hr ft °F were used in the calculations.

The computer program was run, with the use of the data given above, for a continuous firing burst of 12.5 seconds. All portions of the main program and its subroutines functioned properly. No numerical instability nor convergence problems were encountered. The temperatures at all nodes were printed at time intervals of approximately one second. The computed bore surface temperatures, as functions of time and position, are shown graphically in Figure 3. The bore surface temperature rises more rapidly at locations in which the barrel wall is thinnest, as would be expected. The effects of axial temperature gradients are minor, except where an abrupt change occurs in the external diameter. This indicates that, for this specific example, a less complicated and less expensive one-dimensional numerical program could be used over most of the axial length without the introduction of any major errors. The two-dimensional program could still be used in regions where a step change exists in diameter.

A summary of this investigation is given in the following section.

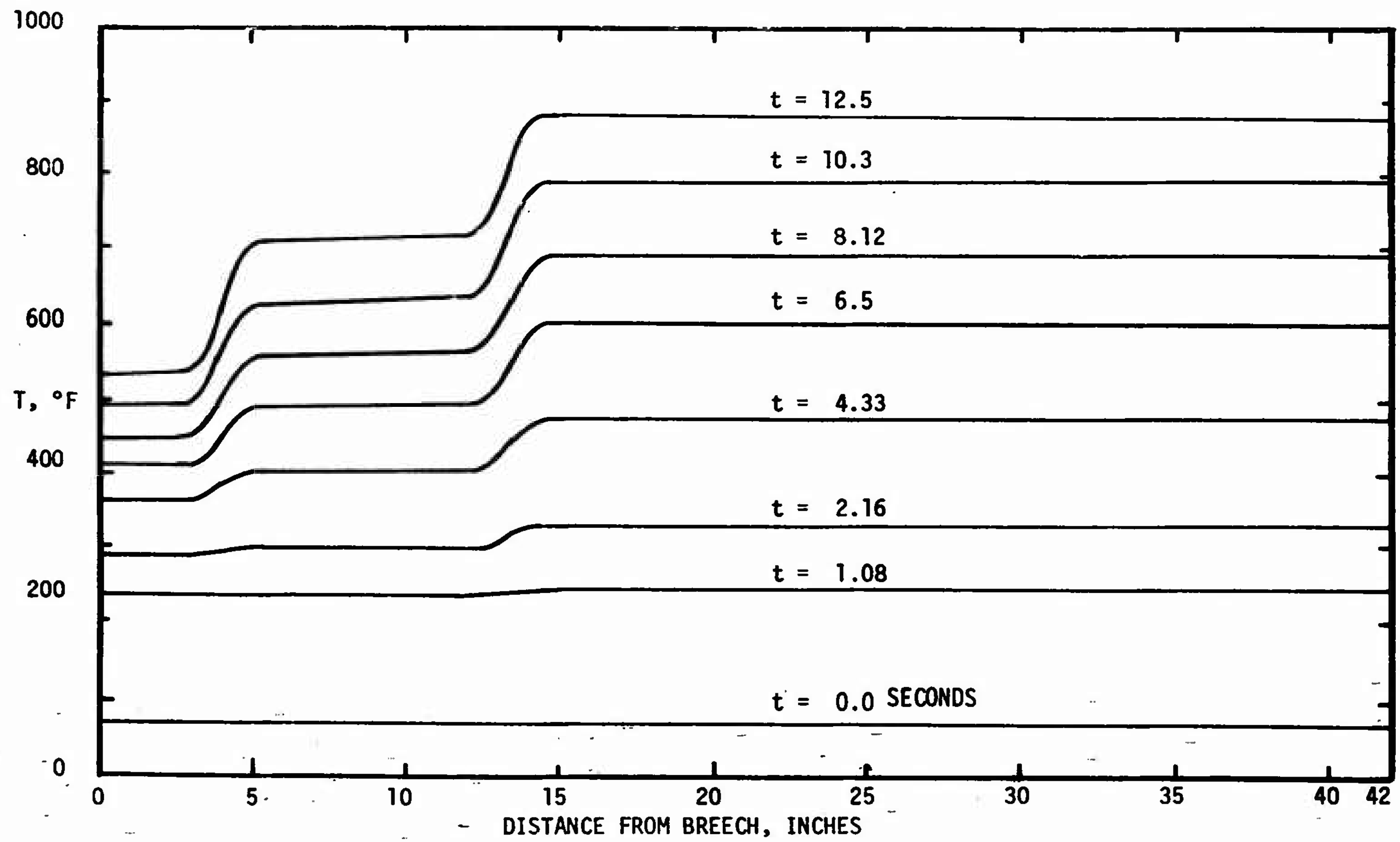


FIGURE 3 BORE TEMPERATURE VERSUS DISTANCE FROM BREECH AS A FUNCTION OF TIME

SUMMARY

A mathematical model to determine the two-dimensional temperature profile in a gun tube, under realistic physical conditions, is presented. Variable geometry, temperature dependent thermal properties, and variable conditions at the boundaries were considered in the mathematical model. A numerical algorithm was developed for the mathematical model by use of the method of explicit finite differences. The numerical algorithm was programmed for evaluation by the digital computer. A numerical example was computed to check the computer program. The program and all its subroutines functioned properly. No numerical instability nor convergence problems were encountered.

LITERATURE CITED

1. Dusinberre, G. M., Numerical Analysis of Heat Flow, 1st Edition, McGraw-Hill, New York, 1949.

2. Aerospace Structural Metals Handbook, Volume 1, "Ferrous Alloys," Third Revision, Syracuse University Press, March 1966.

APPENDIX A

DIGITAL COMPUTER PROGRAM

```
C          A DIGITAL COMPUTER PROGRAM TO DETERMINE TWO-DIMENSIONAL TEMP-
C            ERATURE PROFILE IN GUN TUBES BY THE METHOD OF EXPLICIT
C              FINITE DIFFERENCES.
C
      DIMENSION T(40,30), QR(40,30), QZ(40,30), DR(5), DZ(5), HG(40),
     1  TG(40),  R(40), CP(40), RHO(40), EMIS(40), XMC(40), XMR(40),
     2  DT(50), DTR(40,30), DTR2(40,30), DTZ(40,30), DTZ2(40,30),
     3  JJS(8), LLS(8), RS(9), ZS(8), XJS(8), XLS(8), JLL(8), KRA(8),
     4  X(20), Y(20)
      COMMON /BLK1/JSUM(8), N, NKONT, NP, ISUM, LL1, M, SIGMA, TS
      READ 1, M, N, SIGMA, TS, TTIME, NP, NR
      PRINT 201, M, N, SIGMA, TS, TTIME, NP
  201 FORMAT(4H M =, I5, 5X, 4H N =, I5, 5X, 8H SIGMA =, E15.6, 5X,
     1 5H TS =, E12.5, 5X, 8H TTIME =, E12.5, 5X,5H NP =, I8)
C          NP  - WHICH ITERATION DESIRE DERIVATIVE PRINTS.
C          NR - MODULUS NR INDICATES WHICH ITERATIONS TEMP PRINT OUTS ARE DESIRED.
      READ 21,(JJS(I), I = 1, M)
      PRINT 202, (JJS(I), I = 1,M)
  202 FORMAT(4H JJS/(7I15))
      READ 21,(LLS(J), J = 1, N)
      PRINT 203, (LLS(J), J =1,N)
  203 FORMAT(4H LLS/(7I15))
      MP1 = M + 1
      READ 2, ( RS(J), J = 1, MP1)
      PRINT 204, (RS(J), J = 1,MP1)
  204 FORMAT(3H RS/ (7E15.7))
C          ALL SEGMENTS K SEGMENTS OUT RADIALLY MUST HAVE SAME RADIAL DIMENSION.
      READ 2, ( ZS(I), I = 1, N)
      PRINT 205,  (ZS(I), I = 1,N)
  205 FORMAT(4H  ZS/ (7E15.7))
C          ALL SEGMENTS K SEGMENTS OUT AXIALLY  MUST HAVE SAME AXIAL  DIMENSION.
      READ 21,(KRA(I), I = 1, N)
      PRINT 206, (KRA(I), I = 1,N)
  206 FORMAT(4H KRA/ (7I15))
      READ 2, ( X(I), Y(I), I = 1,10)
    1 FORMAT(2I10, E15.5, 2F10.0,2I10)
    2 FORMAT(8F10.0)
   21 FORMAT(8I10)
C
C     PRINT 330,KRA(1), KRA(2), KRA(3)
C 330 FORMAT(9H KRA(1) =,I4,10X,9H KRA(2) =, I4,10X, 9H KRA(3) =, I4)
C
      KINT = M + N + 1
C     M MUST BE LESS THAN OR EQUAL TO N
C          MUST HAVE N NUMBER OF JSUM(I)'S.
C          KRA(I) IS NUMBER OF RADIAL SECTIONS PER EACH AXIAL SECTION.
      DO 3 K=1,N
```

```
      KRAD = KRA(K)
      JSUM(K) = 0
      DO 3 J = 1,KRAD
    3 JSUM(K) = JJS(J) + JSUM(K)
C
      PRINT 332,JSUM(1), JSUM(2), JSUM(3)
  332 FORMAT(10H JSUM(1) =,I4,10X,10H JSUM(2) =,I4,10X,10H JSUM(3) =,I4)
C
      ISUM = 0
      DO 4 I = 1, N
    4 ISUM = ISUM + LLS(I)
C          ISUM REPRESENTS THE TOTAL NUMBER OF AXIAL NODES
C          JJS- NUMBER OF RADIAL NODES IN THE RESPECTIVE SEGMENTS.
C          LLS- NUMBER OF AXIAL NODES IN THE RESPECTIVE SEGMENTS.
C            N - NUMBER OF AXIAL SEGMENTS.
C            M - MAXIMUM NUMBER OF RADIAL SEGMENTS.
C          T(I,J) - TEMPERATURES.
C          DR(I) - RADIAL INCREMENT CHANGE (DR(1) < DR(2) < DR(3) )
C          DZ(I) - AXIAL INCREMENT CHANGE (START WITH DZ(1) = DZ(2) = DZ(3) )
C            TRIAL AND ERROR FOR CORRECT VALUES FOR DZ(I)'S.
C          RS(J)  -  BOUNDARY RADII FROM BORE     TO OUTSIDE.
C          ZS(I) - - AXIAL BOUNDARIES.
C          TS - AMBIENT TEMPERATURE.
C          HG(IT) - WHERE IT IS TEMP SUBSCRIPT AND HG IS THE CONVECTION COEFFICIEN
C              T OF THE GAS COMPUTED AS A FUNCTION OF TEMP.
C          XK     -  THERMAL CONDUCTIVITY AS A FUNCTION OF TEMP.
C          CP(IT) - SPECIFIC HEAT AS A FUNCTION OF TEMP.
C          RHO(IT) - DENSITY AS A FUNCTION OF TEMP.
C          EMIS(IT) - EMISSIVITY AS A FUNCTION OF TEMP.
C          XHC(IT) - CONVECTION COEFFICIENT AS A FUNCTION OF TEMP.
C          XHR(IT) - RADIATION AS A FUNCTION OF TEMP.
C          TTIME IS TIME OF TERMINATION
C
C    .     THE FOLLOWING COMPUTES THE AXIAL AND RADIAL CHANGES (DELTAS)
      DO 6 J = 1, M
    6 XJS(J) = JJS(J)
      DO 7 I = 1, N
    7 XLS(I) = LLS(I)
      DZ(1) = ZS(1) / (XLS(1) - 1.0)
      DO 8 I= 2, N
    8 DZ(I) = (ZS(I) - ZS(I-1))/ XLS(I)
C          NEXT COMPUTE THE RADII
      DR(1) = (RS(2) - RS(1)) / (XJS(1) - 1.0)
      R(1) = RS(1)
      JKK = 2
      JJJ = 0
      DO 11 JK= 1, M
      IF(JK .EQ. 1) GO TO 9
      DR(JK) = (RS(JK+1) - RS(JK)) / XJS(JK)
      JKK = JKK + JJS(JK-1)
    9 JJJ = JJJ + JJS(JK)
      DO 10 J = JKK, JJJ
   10 R(J) = R(J-1) + DR(JK)
      J= J + 1
      R(J) = RS(JK + 1)
   11 CONTINUE
      R(J) = RS(M+1)
      TIME = 0.0
      NKONT = 1
      TSAB = TS + 460.0
      NN = 0
```

```
      JI1(1) = JJS(1) + 1
      MM1 = M-1
      DO 14  J = 2,MM1
   14 JI1(J) = JI1(J-1) + JJS(J)
      IK = N - LLS(M)
      IK = 1
      NN = LLS(1)
      DO 18  II = 1,M
      JSU = JSUM(II)
      DO 17  I = IK,NN
      DO 17  J = 1, JSU
   17 T(I,J) = TS
      IF(II .EQ. M) GO TO 18
      IK = IK + LLS(II)
      NN = NN + LLS(II+1)
   18 CONTINUE
   60 CONTINUE
C
C          NEXT COMPUTE THE MAXIMUM TIME INTERVAL.
C     DX = DR(1)                (SINCE DR(1) < DR(I) OR DZ(I) )
      DX = DR(1)
C
C          NOW LOCATE MAXIMUM XK(IT), AND MINIMUM XM(IT)
C     XKMAX = XK(IT)MAX
C     XMMINM= XM(IT)MIN
      XKMAX = 28.3
      XKMIN = 12.8
      XMMAX = 240.0
      XMMIN = 207.0
C
      XMR     = 4. + 4. * XMMAX / XKMIN * DX
C          NOW LOCATE RHO(IT) MINIMUM, CP(IT) MINIMUM.
C     RHOMIN = RHO(IT)MIN
C     CPMIN = CP(IT)MIN
      RHOMIN = 490.0
      CPMIN = .108
C
      DT(1) = RHOMIN * CPMIN / XMR     / XKMAX * DX * DX
      XMR  = 4. + 4. * XMMIN / XKMAX * (RS(2) - RS(1)) / (XLS(1) - 1.0)
      DT(2) = RHOMIN * CPMIN / XMR / XKMAX * (RS(2)-RS(1))**2 / XLS(1)   -1.0
     1       ** 2
      DO 19 J = 2, M
      XMR = 4. + 4. * XMMIN / XKMAX * (RS(J+1) - RS(J)) / XLS(J)
   19 DT(J+1) = RHOMIN * CPMIN / XMR / XKMAX * (RS(J+1) - RS(J)) ** 2 /
     1   XLS(J) ** 2
      XMZ  = 4. + 4. * XMMIN / XKMAX * ZS(1) / (XJS(1) - 1.0 )
      DT(M+1) = RHOMIN * CPMIN / XMZ  / XKMAX * ZS(1) ** 2 / XJS(1) ** 2
      DO 190 I = 2, M
      XMZ = 4. + 4. * XMMIN / XKMAX * (ZS(I) - ZS(I-1)) / XJS(I)
  190 DT(I+M)    = RHOMIN * CPMIN / XMZ / XKMAX * (ZS(I) - ZS(I-1)) ** 2
     1   /  XJS(I) **2
C          NOW MUST DETERMINE SMALLEST DT AND SAID VALUE WILL BE THE TIME
C                    INCREMENT.
      IDT = 0
   20 IDT = IDT +1
      IPT = IDT + 1
      IF(DT(IDT) .GT. DT(IPT)) GO TO 30
      DT(IPT) = DT(IDT)
   30 KINT1 = KINT - 1
      IF(IPT .NE. KINT1) GO TO 20
C            KINT MUST BE READ IN AS (N + M + 1)
```

```
      DTH = DT(KINT1)
      DTH = DTH / 1.5
      TIME = TIME + DTH * 3600.0
C        NOW COMPUTE ALL VALUES DEPENDENT ON TEMPERATURE
C        SUBSCRIPT IT DESIGNATES WHICH TIME INTERVAL
C        XHC(I,J) MUST EITHER BE A TABLE READ IN OR A FUNCTION COMPUTATION.
C        NEXT COMPUTE THE HEAT FLUXES.
      LL1 = LLS(1)
      CALL QRSUB(T,QR,TSAB,JM,LLS)
C                      *     *     *     *     *     *     *
      CALL QZSUB(TSAB,JJS,LLS,T,QZ,KRA)
C        NOW CHECK RADIAL TEMP AGAINST AXIAL TEMP DIFF AND SHOULD BE CLOSE.
C           ONLY NEED TO CHECK BORE TEMPS AXIALLY AGAINST ADJACENT RADIAL TEMPS.
      CALL RADDER(T,DR,DTR,DTR2,QR,JJS,LLS)
C        NEXT COMPUTE AXIAL DERIVATIVES.
      CALL AXIDER(T,DZ,DTZ,DTZ2,QZ,LLS)
C           COMPUTE TEMP CHANGE AT EACH NODE
      IF(NKONT .EQ. 1) GO TO 490
      IF(MOD(NKONT,NR) .NE. 0) GO TO 510
C
  490 PRINT 500, TIME, NKONT
  500 FORMAT(//5X, 8H TIME = , E12.4, 10X, 18H ITERATION NUMBER , I5/)
  510 IFF = 0
      DO 503 IK =1, N
      IIN = IFF + 1
      IFF = IFF + LLS(IK)
      JSU = JSUM(IK)
      DO 434 J = 1, JSU
      DO 434 I = IIN, IFF
      TT = T(I,J)
      CALL DKDT(TT, XKK, DKT, XRHO, XCP,X, Y)
C           DKDT IS THE THERMAL CONDUCTIVITY , DENSITY, & SPECIFIC HEAT SUBROUTIN
      IF(NKONT .NE. 0) GO TO 803
      PRINT 802, XKK, XRHO, XCP, DTR2(I,J), R(J), DTR(I,J), DTZ2(I,J),
     1 DKT, DTZ(I,J)
  802 FORMAT(6H XKK =, E15.7, 5X, 7H XRHO =, E15.7, 5X, 6H XCP =, E15.7,
     1 5X, 12H DTR2(I,J) =, E15.7 / 7H R(J) =, E15.7, 5X, 11H DTR(I,J)
     2=, E15.7,5X,12H DTZ2(I,J) =, E15.7, 5X, 6H DKT =, E15.7 / 5X, 12H
     3 DTZ(I,J) =, E15.7)
  803 DTOT = XKK / XRHO / XCP * (DTR2(I,J) + 1. /     (R(J))     * DTR(I,
     1  J) + DTZ2(I,J)) + 1. / XRHO / XCP * DKT * (DTR(I,J) ** 2 +
     2   DTZ(I,J) ** 2)
  434 T(I,J) = T(I,J) + DTOT * DTH
      IF(NKONT .EQ. 1) GO TO 441
      IF(MOD(NKONT,NR) .NE. 0) GO TO 503
C
  441 CONTINUE
      DO 501 I = IIN, IFF
      PRINT 502, I
  502 FORMAT(5X, 18H AXIAL LOCATION - , I5)
      PRINT 505, (T(I,J), J = 1,JSU)
  501 CONTINUE
  503 CONTINUE
  505 FORMAT(8(3X,E12.4))
  560 NKONT = NKONT + 1
C     IF(NKONT .LE. 500) GO TO 60
      IF(TIME .LT. TTIME) GO TO 60
  760 CALL EXIT
      END
```

```
      SUBROUTINE CONV(TT,XXHC,EMISS,XK)
      EMISS = .5
      XXHC = 5.0
      CALL XKKS(TT,XK,DKT)
      RETURN
      END


      SUBROUTINE QZSUB(TSAB,JJS,LLS,T,QZ,KRA)
      COMMON /BLK1/JSUM(8), N, NKONT, NP, ISUM, LL1, M, SIGMA, TS
      DIMENSION QZ(40,30), T(40,30), JJS(8), LLS(8), KRA(8), X(20),
     1  Y(20)
      DO 90 KR=1,N
      ISU = 0
      DO 5 IK = 1, KR
    5 ISU = ISU + LLS(IK)
      IF(KR .NE. N) GO TO 8
      JSU = JSUM(N)
      DO 12 J = 1,JSU
      TABS = T(ISUM,J) + 460.0
      TT = T(ISUM,J)
      CALL CONV(TT, XXHC, EMISS, XK)
      XXHR = EMISS * SIGMA      * (TABS ** 3 + TABS ** 2 * (TSAB) +
     1   TABS * (TSAB)         ** 2 + (TSAB)          ** 3)
      XXH = XXHC   + XXHR
      IF(NKONT .NE. 0) GO TO 6
      PRINT 201, XK, XXHR, XXH
  201 FORMAT(5H XK =, E15.7, 5X, 7H XXHR =, E15.7, 5X, 6H XXH =, E15.7)
    6 QZ(ISUM,J) = XXH / XK       * (T(ISUM,J) - TS)
      IF(NKONT .NE. 0) GO TO 12
      PRINT 10, ISUM, J, QZ(ISUM,J)
   10 FORMAT(5X,9H, ISUM = , I5,5X,3H J=,I5, 10X, 11H QZ(M,J) = , E12.4)
   12 CONTINUE
      GO TO 90
    8 JSU = JSUM(KR)
      JSU1 = JSUM(KR+1)
      IF(JSU .EQ. JSU1) GO TO 90
      IF(JSU .LT. JSU1) GO TO 80
      JSU11 = JSU1 + 1
      DO 70 J = JSU11, JSU
      TABS = T(ISU,J) + 460.0
      TT = T(ISU, J)
      CALL CONV(TT, XXHC, EMISS, XK)
      XXHR = EMISS * SIGMA      * (TABS ** 3 + TABS ** 2 * (TSAB) +
     1   TABS * (TSAB)         ** 2 + (TSAB)          ** 3)
      XXH = XXHC   + XXHR
      IF(NKONT .NE. 0) GO TO 9
      PRINT 201, XK, XXHR, XXH
    9 QZ(ISU, J) = XXH / XK       * (T(ISU, J) - TS)
      IF(NKONT .NE. 0) GO TO 70
      PRINT 203, ISU, J, QZ(ISU,J)
  203 FORMAT(6H ISU =, I10, 5X, 4H J =, I10, 5X, 12H QZ(ISU,J) =, E15.7)
   70 CONTINUE
      GO TO 90
   80 JSUM1 = JSU + 1
      ISU1 = ISU + 1
      DO 85 J = JSUM1, JSU1
      TABS = T(ISU1,J) + 460.0
      TT = T(ISU1,J)
```

```
      CALL CONV(TT, XXHC, EMISS, XK)
      XXHR = EMISS * SIGMA      * (TABS ** 3 + TABS ** 2 * (TSAB) +
     1   TABS * (TSAB)          ** 2 + (TSAB)          ** 3)
      XXH = XXHC  + XXHR
      IF(NKONT .NE. 0) GO TO 85
      PRINT 201, XK
      QZ(ISU1,J) = XXH / XK      * (TS - T(ISU1,J))
   85 CONTINUE
   90 CONTINUE
      RETURN
      END


      SUBROUTINE QRSUB(T,QR,TSAB,JM,LLS)
      COMMON /BLK1/JSUM(8), N, NKONT, NP, ISUM, LL1, M, SIGMA, TS
      DIMENSION T(40,30), QR(40,30),LLS(8), X(20), Y(20)
      DO 20 I=1,ISUM
      TABS = T(I,1) + 460.0
      TT = T(I,1)
      CALL CONV(TT,XXHC,EMISS,XK)
      XXHR = EMISS * SIGMA      * (TABS ** 3 + TABS ** 2 * (TSAB) +
     1   TABS * (TSAB)          ** 2 + (TSAB)          ** 3)
      XXH = XXHC  + XXHR
   20 QR(I,1) = XXH / XK * (T(I,1) - TS)
      LI = LL1
      IS = 0
      DO 200 IK =1,N
      JSU = JSUM(IK)
      II = IS +1
      IS = IS + LLS(IK)
      DO 130 I = II, IS
      TABS = T(I,JSU) + 460.0
      TT = T(I,JSU)
      CALL CONV(TT, XXHC, EMISS, XK)
      XXHR = EMISS * SIGMA      * (TABS ** 3 + TABS ** 2 * (TSAB) +
     1   TABS * (TSAB)          ** 2 + (TSAB)          ** 3)
      XXH = XXHC  + XXHR
  130 QR(I,JSU)= XXH / XK  * (T(I,JSU)- TS)
  200 CONTINUE
      RETURN
      END


      SUBROUTINE AXIDER(T,DZ,DTZ,DTZ2,QZ,LLS)
      DIMENSION T(40,30), DTZ(40,30), DTZ2(40,30), DZ(5), QZ(40,30),
     1  LLS(8), X(20), Y(20)
      COMMON /BLK1/JSUM(8), N, NKONT, NP, ISUM, LL1, M, SIGMA, TS
      IS = 0
      DO 320 IK =1, N
      JSU = JSUM(IK)
C     IF(IK .EQ. N) GO TO 150
C     IF(JSU .LE. JSUM(IK+1)) GO TO 150
      II = IS +1
      IS = IS + LLS(IK)
  150 DO 320 J=1,JSU
      IF(II .NE. 1) GO TO 185
      QZ(1,J)=0.0
      DTZ(1,J) =-QZ(1,J)
      DTZ2(1,J) = 1./18./DZ(1) ** 2 *(-85. * T(1,J) + 108.0 * T(2,J) -
```

```
    1  27.0 * T(3,J) + 4.0 * T(4,J)) + 11. / 3.0 * QZ(1,J) / DZ(1)
      IF(MOD(NKONT,NP) .NE. 0) GO TO 152
      PRINT 20, J, DTZ(1,J), DTZ2(1,J)
   20 FORMAT(7H DTZ(1,,I3, 3H) =, E15.6, 10X, 7H DTZ2 =,E15.6)
  152 I2 = II   1
      DTZ(I2,    1./6./DZ(IK) * ( -2. * T(I2-1,J) - 3. * T(I2,J) + 6. *T(
    1I2,J)    ''I2+2,J))
      DTZ2(I2,J) = 1. /DZ(IK) ** 2 * (T(I2+1,J) - 2. * T(I2,J) + T(I2-1,
    1  J))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 31
      PRINT 22, I2, J, DTZ(I2,J), DTZ2(I2,J)
   22 FORMAT(5H DTZ(,I3,2H , I3, 3H) =, E15.6, 10X, 7H DTZ2 =, E15.6)
   31 INT = II + 2
      IS2 = IS - 2
      DO 153 I = INT, IS2
      DTZ(I,J) = 1. / 12. / DZ(IK)* (T(I - 2, J) - 8. * T(I - 1, J) + 8.
    1* T(I + 1, J) - T(I + 2, J))
C 153 DTZ2(I,J) = 1. / 12. / DZ(IK)** 2 * (-T(I-2,J) + 16. * T(I-1,J) -
      DTZ2(I,J) = 1. / 12. / DZ(IK)** 2 * (-T(I-2,J) + 16. * T(I-1,J) -
    1  30. * T(I,J) + 16. * T(I+1,J) - T(I+2,J))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 153
      PRI'''  '2, I,   J, DTZ(I, J), DTZ2(I, J)
  1'3 CONTINUE
      III = IS2 + 1
      DTZ(III,J)= 1. / 6. / DZ(IK)* ( -6. * T(III-1,J)+ 3. * T(III,J)+ 2
    1.*T(III+1,J) + T(III-2,J))
      DTZ2(III,J)= 1. / DZ(1) ** 2 * (T(III+1,J) - 2. * T(III,J)+T(III-1
    1,J))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 32
      PRINT 22, III,J, DTZ(III,J),DTZ2(III,J)
   32 IF(IK .EQ. N) GO TO 165
      IF(J .GT. JSUM(IK+1)) GO TO 165
      DTZ(IS,J) = 1./DZ(IK+1)/ (1.+DZ(IK+1)/ DZ(IK))* ( T(IS+1, J) - T(I
    1S,J) * (1.-DZ(IK+1)** 2 / DZ(IK)** 2)-DZ(IK+1)** 2 / DZ(IK)** 2
    2  * T(IS-1,J))
      DTZ2(IS,J) = 2./(DZ(IK)*DZ(IK+1)+DZ(IK+1)**2)*(DZ(IK+1)/DZ(IK)*
    1  T(IS-1,J)-(DZ(IK+1)/ DZ(IK)+ 1.) * T(IS,J) + T(IS+1,J))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 320
      PRINT 22, IS, J, DTZ(IS,J), DTZ2(IS,J)
      GO TO 320
  160 IF(IK .EQ. 1) GO TO 320
      IF(J .LE. JSUM(IK-1)) GO TO 185
  165 DTZ(IS,J) =-QZ(IS,J)
      DTZ2(IS,J) = 1. / 18. / DZ(IK)** 2 * (-85. * T(IS,J) + 108. * T(IS
    1-1,J) - 27. * T(IS-2,J) + 4. * T(IS-3,J)) - 11. * QZ(IS,J) / 3. /
    2  DZ(IK)
      IF(MOD(NKONT,NP) .NE. 0) GO TO 320
      PRINT 22, IS, J, DTZ(IS,J), DTZ2(IS,J)
      GO TO 320
  185 IF(JSU .GT. JSUM(IK-1)) GO TO 200
  190 DTZ(II,J) = 1. / 6. / DZ(IK)* (-2. * T(II-1,J) - 3. * T(II,J)+ 6.*
    1  T(II+1,J) - T(II+2,J))
      DTZ2(II,J) = 1./DZ(2) ** 2 * (T(II+1,J) - 2. * T(II,J) +T(II-1,J))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 33
      PRINT 22, II, J, DTZ(II,J), DTZ2(II,J)
   33 GO TO 152
  200 IF(J .LE. JSUM(IK-1)) GO TO 190
      DTZ(II,J) = QZ(II,J)
      DTZ2(II,J) = 1./18./DZ(IK)** 2 *(-85. * T(II,J) + 108.0 * T(II+1,J
    1)-27.0 * T(II+2,J) + 4.0 * T(II+3,J)) - 11. * 3.0 * QZ(II,J) / DZ(
    2 IK)
```

```
      IF(MOD(NKONT,NP) .NE. 0) GO TO 34
      PRINT 22, II, J, DTZ(II,J), DTZ2(II,J)
   34 IF(IK .NE. N) GO TO 150
      GO TO 152
  320 CONTINUE
      RETURN
      END


      SUBROUTINE RADDER(T,DR,DTR,DTR2,QR,JJS,LLS)
      DIMENSION T(40,30), DR(5), DTR(40,30), DTR2(40,30), QR(40,30),
     1 JJS(8), LLS(8), X(20), Y(20)
      COMMON /BLK1/JSUM(8), N, NKONT, NP, ISUM, LL1, M, SIGMA, TS
C
      XH = 200.0
      TG = 2000.0
C
      IFT = 1
      ILA = LLS(1)
      DO 360 IK = 1, N
      JSX = JSUM(IK)
      DO 305 JR = 1,M
      JSX = JSX - JJS(JR)
      IF(JSX .EQ. 0) GO TO 306
  305 CONTINUE
      JR = M
  306 CONTINUE
      DO 350 I = IFT,ILA
      TT = T(I,1)
      CALL CONV(TT, XXHC, EMISS, XK)
      QR(I,1)=XH * (TG - T(I,1)) / XK
      DTR(I,1) =-QR(I,1)
C           QR(I,1) = ? ?
      DTR2(I,1) = 1. / 18. / DR(1) ** 2 * (-85. * T(I,1) + 108. * T(I,2)
     1 - 27. * T(I,3) + 4. * T(I,4)) + 11. / 3. / DR(1) * QR(I,1)
      IF(MOD(NKONT,NP) .NE. 0) GO TO 40
      PRINT 20, I, DTR(I,1), DTR2(I,1)
   20 FORMAT(5H DTR(,I3, 5H,1) =, E15.6, 10X, 7H DTR2 =, E15.6)
   40 DTR(I,2) = 1. / 6. / DR(1) * (-2. * T(I,1) - 3. * T(I,2) + 6. *
     1 T(I,3) - T(I,4))
      DTR2(I,2) = 1. / DR(1) ** 2 * (T(I,3) - 2. * T(I,2) + T(I,1))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 41
      PRINT 22, I, DTR(I,2), DTR2(I,2)
   22 FORMAT(5H DTR(,I3, 5H,2) =, E15.6, 10X, 7H DTR2 =, E15.6)
   41 JJ1 = JJS(1) - 2
      DO 310 J = 3, JJ1
      DTR(I,J) = 1. / 12. / DR(1) * (T(I,J-2) - 8. * T(I,J-1) + 8. *
     1 T(I,J+1)  -  T(I,J+2))
      DTR2(I,J) = 1. / 12. / DR(1) ** 2 * (-T(I,J-2) + 16. * T(I,J-1) -
     1 30. * T(I,J) + 16. * T(I,J+1) - T(I,J+2))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 310
      PRINT 24, I, J,  DTR(I,J),  DTR2(I,J)
  310 CONTINUE
   24 FORMAT(5H DTR(,I3,2H ,I3,3H) =, E15.6, 10X, 7H DTR2 =, E15.6)
      DTR(I,JJ1+1) = 1. / 6. / DR(1) * (-6. * T(I,JJ1) + 3. * T(I,JJ1+1)
     1 +2. * T(I,JJ1+2) + T(I,JJ1-1))
      DTR2(I,JJ1+1) = 1. / DR(1) ** 2 * (T(I,JJ1+2) -2. * T(I,JJ1+1)+T(I
     1, JJ1))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 43
      JZ = JJ1 +1
```

```
      PRINT 24, I, JZ, DTR(I,JZ), DTR2(I,JZ)
   43 IF(JJS(1) .EQ. JSUM(IK)) GO TO 325
      JJ = JJS(1)
      DTR(I,JJ) = 1. / DR(2) / (1. + DR(2) / DR(1)) * (T(I,JJ+1) - T(I,
     1  JJ) *(1. - DR(2) ** 2 / DR(1) ** 2) - DR(2) ** 2 / DR(1) ** 2 *
     2  T(I,JJ-1))
      DTR2(I,JJ) = 2. / (DR(1) * DR(2) + DR(2) ** 2) * (DR(2) / DR(1) *
     1 T(I,JJ-1) - (DR(2) / DR(1) + 1) * T(I,JJ) + T(I,JJ +1))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 44
      PRINT 24, I, JJ, DTR(I,JJ), DTR2(I,JJ)
   44 JSX = 0
      DO 315 JK = 2, JR
      JSX = JSX + JJS(JK-1)
      JP = JSX + 1
      DTR(I,JP) = 1. / 6. / DR(JK)* (-2. * T(I,JP-1)-3. * T(I,JP) + 6. *
     1  T(I,JP+1) - T(I,JP+2))
      DTR2(I,JP) = 1. / DR(JK)** 2 * (T(I,JP+1) - 2. * T(I,JP) +T(I,JP-1
     1))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 45
      PRINT 24, I, JP, DTR(I,JP), DTR2(I,JP)
   45 JI = JP + 1
      JL = JI + JJS(JK) - 4
      DO 312 J = JI,JL
      DTR(I,J) = 1. / 12. / DR(JK)* (T(I,J-2) - 8. * T(I,J-1) + 8. *
     1  T(I,J+1) - T(I,J+2))
      DTR2(I,J) = 1. / 12. / DR(JK)** 2 * (-T(I,J-2) + 16. * T(I,J-1) -
     1  30. * T(I,J) + 16. * T(I,J+1) - T(I,J+2))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 312
      PRINT 24, I, J,  DTR(I,J),  DTR2(I,J)
  312 CONTINUE
      JPP = JL + 1
      DTR(I,JPP) = 1./ 6. / DR(JK)* (-6. * T(I,JPP-1) + 3. * T(I,JPP) +
     1  2. * T(I,JPP+1) + T(I,JPP-2))
      DTR2(I,JPP) = 1. / DR(JK)** 2 * (T(I,JPP+1) - 2. * T(I,JPP) + T(I,
     1  JPP-1))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 47
      PRINT 24, I, JPP,DTR(I,JPP),DTR2(I,JPP)
   47 IF(JK .EQ.  JR) GO TO 315
      JPL = JPP + 1
      JZ = JK + 1
      DTR(I,JPL)= 1. / DR(JZ)/ (1. + DR(JZ)/ DR(JK))* (T(I,JPL+1)- T(I,
     1  JPL)*(1. - DR(JZ)** 2 / DR(JK)** 2) - DR(JZ)** 2 / DR(JK)** 2 *
     2  T(I,JPL-1))
      DTR2(I,JPL)= 2. / (DR(JK)* DR(JZ)+ DR(JZ)** 2) * (DR(JZ)/ DR(JK)*
     1 T(I,JPL-1)- (DR(JZ)/ DR(JK)+ 1) * T(I,JPL)+ T(I,JPL+1))
      IF(MOD(NKONT,NP) .NE. 0) GO TO 315
      PRINT 24, I, JPL,DTR(I,JPL),DTR2(I,JPL)
  315 CONTINUE
  325 JLT = JSUM(IK) - 2
      DTR(I,JLT+2) =-QR(I,JLT+2)
      DTR2(I,JLT+2) = 1. / 18./DR(JR)** 2 * (-85. * T(I,JLT+2) + 108. *
     1 T(I, JLT+1) - 27. * T(I, JLT) + 4. * T(I,JLT-1)) - 11. / 3. / DR(
     2JR)* QR(I, JLT+2)
      IF(MOD(NKONT,NP) .NE. 0) GO TO 350
      JW = JLT + 2
      PRINT 24, I, JW, DTR(I,JW), DTR2(I,JW)
  350 CONTINUE
      IF(IK .EQ. N) GO TO 360
      IUN = IK + 1
      IFT = IFT + LLS(IK)
      ILA = ILA + LLS(IUN)
```

```
  360 CONTINUE
      RETURN
      END


      SUBROUTINE DKDT(TT,XK,DKT,XRHO,XCP, X, Y)
      DIMENSION X(20), Y(20)
      CALL XKKS(TT,XK,DKT)
      XRHO = 490.0
      I = 1
      CALL LINEAR(TT,X,Y,XCP,I)
      RETURN
      END


      SUBROUTINE XKKS(TT,XK,DKT)
      IF(TT .GT. 1472.0) GO TO 14
      XK=28.30-.00870*TT
      DKT = - 0.0087
      GO TO 20
   14 XK=10.39+.00347*TT
      DKT =  0.00347
   20 CONTINUE
      RETURN
      END


      SUBROUTINE LINEAR(A,X,Y,VV,I)
      DIMENSION X(20),Y(20)
    1 IF(Y(I+1) .LT. Y(I)) GO TO 100
C        USE FOLLOWING IF AS Y INCREASES X INCREASES
   10 IF(A-X(I))3,2,2
C        USE FOLLOWING IF AS Y INCREASES X DECREASES
  100 IF(A-X(I))2,2,3
    2 I=I+1
      GO TO 1
    3 I=I-1
      VV=Y(I)*(A-X(I+1))/(X(I)-X(I+1))+Y(I+1)*(A-X(I))/(X(I+1)-X(I))
      RETURN
      END


         3         4        .1714E-08  70.0     20.0             99550        200
         7         5         5
         5         5         5         8
.052083   .0703125  .078125   .09896
.26333    1.0417    1.81907   3.50
         3         2         1         1
0.0       .108      200.0     .112      400.0     .125      600.0     .132
800.0     .150      1000.0    .160      1200.0    .185      1600.0    .180
2000.0    .180      2200.0    .150
```

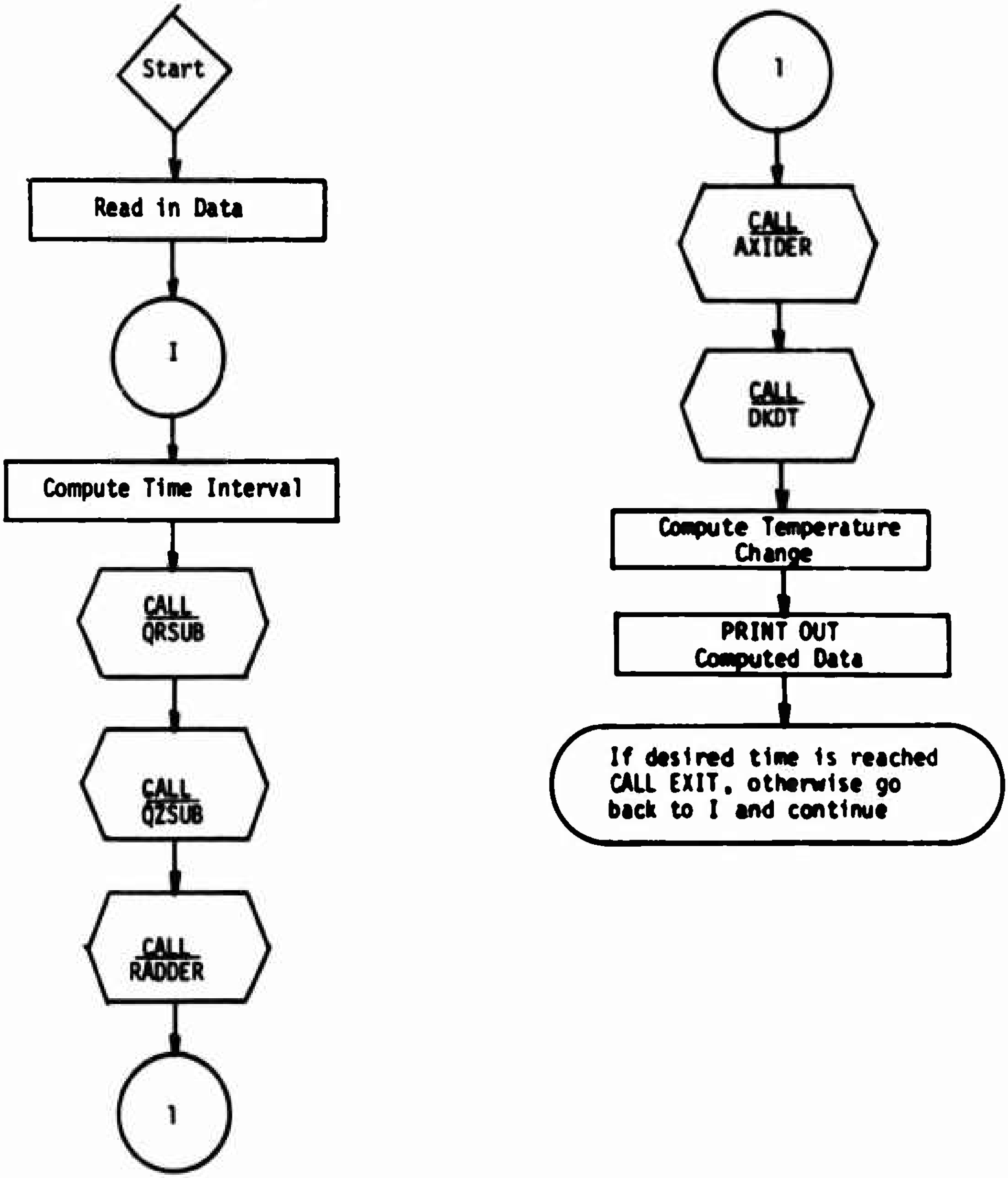
```
M =     3      N =     4        SIGMA =    0.171400E-08      TS = 0.70000E 02        TTIME = 0.20000E 02       VP =   99550
JJS
               7                  5                  5
LLS
               5                  5                  5                  8
RS
 0.5208300E-01  0.7031250E-01  0.7812500E-01  0.9895998E-01
 ZS
 0.2633300E 00  0.1041699E 01  0.1819070E 01  0.3500000E 01
KRA
               3                  2                  1                  1
JSUM(1) =  17          JSUM(2) =  12          JSUM(3) =   7

     TIME =    0.2708E-02                 ITERATION NUMBER      1

     AXIAL LOCATION -        1
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02
     AXIAL LOCATION -        2
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02
     AXIAL LOCATION -        3
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02
     AXIAL LOCATION -        4
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02
     AXIAL LOCATION -        5
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02
     AXIAL LOCATION -        6
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -        7
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -        8
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -        9
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       10
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
   0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       11
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       12
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       13
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       14
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       15


   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       16
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       17
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       18
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       19
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       20
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       21
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       22
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
     AXIAL LOCATION -       23
   0.7641E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02     0.7000E 02
```

```
TIME =    0.1083E 02          ITERATION NUMBER  4000

 AXIAL LOCATION -      1
0.5058E 03   0.4699E 03   0.4381E 03   0.4098E 03   0.3848E 03   0.3627E 03   0.3433E 03   0.3343E 03
0.3259E 03   0.3182E 03   0.3110E 03   0.3045E 03   0.2896E 03   0.2785E 03   0.2707E 03   0.2661E 03
0.2644E 03
 AXIAL LOCATION -      2
0.5058E 03   0.4699E 03   0.4381E 03   0.4098E 03   0.3847E 03   0.3627E 03   0.3433E 03   0.3343E 03
0.3259E 03   0.3182E 03   0.3110E 03   0.3045E 03   0.2896E 03   0.2785E 03   0.2707E 03   0.2661E 03
0.2644E 03
 AXIAL LOCATION -      3
0.5058E 03   0.4699E 03   0.4381E 03   0.4098E 03   0.3847E 03   0.3627E 03   0.3433E 03   0.3343E 03
0.3259E 03   0.3182E 03   0.3110E 03   0.3045E 03   0.2896E 03   0.2785E 03   0.2707E 03   0.2661E 03
0.2644E 03
 AXIAL LOCATION -      4
0.5060E 03   0.4702E 03   0.4383E 03   0.4100E 03   0.3850E 03   0.3629E 03   0.3435E 03   0.3345E 03
0.3261E 03   0.3184E 03   0.3112E 03   0.3046E 03   0.2898E 03   0.2786E 03   0.2708E 03   0.2662E 03
0.2646E 03
 AXIAL LOCATION -      5
0.5091E 03   0.4713E 03   0.4414E 03   0.4130E 03   0.3879E 03   0.3657E 03   0.3462E 03   0.3371E 03
0.3286E 03   0.3207E 03   0.3135E 03   0.3067E 03   0.2914E 03   0.2799E 03   0.2718E 03   0.2671E 03
0.2654E 03
 AXIAL LOCATION -      6
0.6561E 03   0.6229E 03   0.5949E 03   0.5717E 03   0.5530E 03   0.5383E 03   0.5275E 03   0.5233E 03
0.5201E 03   0.5177E 03   0.5162E 03   0.5156E 03
 AXIAL LOCATION -      7
0.6594E 03   0.6262E 03   0.5983E 03   0.5752E 03   0.5565E 03   0.5419E 03   0.5311E 03   0.5269E 03


0.5237E 03   0.5213E 03   0.5198E 03   0.5192E 03
 AXIAL LOCATION -      8
0.6595E 03   0.6263E 03   0.5983E 03   0.5752E 03   0.5565E 03   0.5419E 03   0.5311E 03   0.5270E 03
0.5237E 03   0.5214E 03   0.5199E 03   0.5193E 03
 AXIAL LOCATION -      9
0.6597E 03   0.6264E 03   0.5985E 03   0.5754E 03   0.5567E 03   0.5421E 03   0.5313E 03   0.5271E 03
0.5239E 03   0.5215E 03   0.5201E 03   0.5175E 03
 AXIAL LOCATION -     10
0.6618E 03   0.6286E 03   0.6007E 03   0.5776E 03   0.5588E 03   0.5441E 03   0.5332E 03   0.5290E 03
0.5257E 03   0.5233E 03   0.5218E 03   0.5212E 03
 AXIAL LOCATION -     11
0.8086E 03   0.7780E 03   0.7539E 03   0.7358E 03   0.7230E 03   0.7153E 03   0.7124E 03
 AXIAL LOCATION -     12
0.8117E 03   0.7811E 03   0.7571E 03   0.7389E 03   0.7262E 03   0.7185E 03   0.7156E 03
 AXIAL LOCATION -     13
0.8117E 03   0.7812E 03   0.7571E 03   0.7390E 03   0.7263E 03   0.7186E 03   0.7156E 03
 AXIAL LOCATION -     14
0.8117E 03   0.7812E 03   0.7571E 03   0.7390E 03   0.7263E 03   0.7186E 03   0.7156E 03
 AXIAL LOCATION -     15
0.8117E 03   0.7812E 03   0.7571E 03   0.7390E 03   0.7263E 03   0.7186E 03   0.7156E 03
 AXIAL LOCATION -     16
0.8117E 03   0.7812E 03   0.7571E 03   0.7390E 03   0.7253E 03   0.7186E 03   0.7156E 03
 AXIAL LOCATION -     17
0.8117E 03   0.7812E 03   0.7571E 03   0.7390E 03   0.7261E 03   0.7196E 03   0.7156E 03
 AXIAL LOCATION -     18
0.8117E 03   0.7812E 03   0.7571E 03   0.7390E 03   0.7263E 03   0.7186E 03   0.7156E 03
 AXIAL LOCATION -     19
0.8117E 03   0.7812E 03   0.7571E 03   0.7390E 03   0.7263E 03   0.7186E 03   0.7156E 03
 AXIAL LOCATION -     20
0.8117E 03   0.7812E 03   0.7571E 03   0.7390E 03   0.7263E 03   0.7186E 03   0.7156E 03
 AXIAL LOCATION -     21
0.8117E 03   0.7812E 03   0.7571E 03   0.7390E 03   0.7263E 03   0.7186E 03   0.7156E 03
 AXIAL LOCATION -     22
0.8116E 03   0.7811E 03   0.7570E 03   0.7389E 03   0.7262E 03   0.7185E 03   0.7155E 03
 AXIAL LOCATION -     23
0.8101E 03   0.7796E 03   0.7555E 03   0.7374E 03   0.7247E 03   0.7170E 03   0.7140E 03
```

APPENDIX B

FLOW CHART OF MAIN PROGRAM

FLOW CHART OF
MAIN PROGRAM



LIST OF SYMBOLS USED IN TEXT

Symbols

- C - specific heat (BTU/lb °F)
- h - heat transfer coefficient (BTU/hr ft^2 °F)
- i - number of radial nodes
- j - number of axial nodes
- k - thermal conductivity (BTU/hr ft °F)
- m - axial node
- n - radial node
- q'' - heat flux (BTU/hr ft^2)
- R - radial boundary (ft)
- r - radial coordinate (ft)
- T - temperature (°F)
- t - time (hr)
- Z - axial coordinate (ft)
- ΔR - radial increment (ft)
- ΔZ - axial increment (ft)
- ε - emissivity
- ρ - density (lb/ft^3)
- σ - radiation coefficient (0.1714 BTU/hr ft^2 °R^4)

Subscripts

- g - gas
- i - initial value
- m - node m
- n - node n
- 0 - surroundings
- 1 - boundary 1, segment 1
- 2 - boundary 2, segment 2
- 3 - boundary 3, segment 3
- 4 - boundary 4